DATA PROCESSOR AND IP MODULE FOR DATA PROCESSOR

Field of the invention

The present invention relates to a data processor such as a microprocessor, a DSP (digital signal processor) or the like, which supports a virtual memory management system, and more particularly to techniques for use therein that are applied to memory control having a translation lookaside buffer (TLB) for address translation, and in particular a plurality of TLBs.

Background of the Invention

Typically, the virtual memory management system in a data processor is a function devised to make effective use of the physical memory. When the required memory size for execution of a particular process is less than the available physical memory, it is possible to carry out the process by mapping to the physical memory. However, when the memory size required by the process is larger than the physical memory, the process typically has to be divided so that only selected sub-portions are mapped to the physical memory as needed.

Typically, a system in which the mapping to the physical memory is controlled and carried out as a batch by an OS (Operating System) in such a manner that the process itself is not aware of the mapping to the physical memory is called a virtual memory management system.

In a virtual memory management system, a sufficiently large virtual memory, as compared to the physical memory, is provided to allow the process to be mapped to the virtual memory. Because of this, even when a plurality of processes exist substantially simultaneously, each process operates only on the virtual memory. An MMU (Memory Management Unit), which is ordinarily controlled by the OS, is adopted for the mapping from the virtual memory to the physical memory. The MMU updates the physical memory so that virtual memory necessary for the process is mapped smoothly to the physical memory. The update of the physical memory is carried out between secondary memories. Hereinafter, an address space in the virtual memory will be referred to as virtual address space while an address space in the physical memory will be referred to as physical address space.

Although it is possible to realize the function of the MMU via software only, typically it is not efficient to carry out the translation via software every time the process accesses the physical memory. Accordingly, a translation lookaside buffer (TLB) for performing address translation is provided as part of the hardware to store frequently used address translation information. The TLB can be considered as a cache for the address translation information. When address translation is carried out from the virtual memory to the physical memory using the MMU, if the translation information has not been registered in the TLB, the MMU issues a TLB miss exception and registers new address translation information in the TLB. However, unlike the cache, when the address translation has failed, i.e., when the TLB miss exception is issued, the replacement of the address translation information is ordinarily carried out via software.

It has been determined that problems exist when it is desired to expand the physical memory in a virtual memory management system. As for the virtual memory management system, it was initially assumed that only a relatively small physical address space was desired for the virtual address space from the viewpoint of actual cost and mounting space. However, owing to the tendency of large capacity and low cost of DRAM (Dynamic Random Access Memory) typically used for main memory, it has become possible to allow a physical address space equivalent in size to the typically desired virtual address space. As a result, in certain situations it has become easier to extend the main memory as the physical address space.

Fig. 7 illustrates an example of a conventional approach involving an allocation of a virtual address space to the physical address space. In this example, the virtual address space is defined via 32 bits while the physical address space is defined via 29 bits. The size of the virtual address space depends on the bit length that handles the address. In a 32-bit microprocessor, the virtual address space is an area of 4G bytes (or "4GB"), i.e., the 32nd power of 2. On the other hand, the physical address space depends on the size for mounting actual memory and various bus interfaces. For example, in a microprocessor having a physical address space of 29 bits, the physical address space is an area of 512M bytes (0.5GB); i.e., the 29th power of 2.

A buffer in which is registered address translation information to carry out address translation from the virtual address space to the physical address space is the TLB. In the example of Fig. 7, a portion included in the area P0 (2GB) of the virtual address space is subjected to the address translation into the physical address space of 0.5GB by the TLB. On the other hand, since Pl and P2, which follow P0, are the areas excluded from the TLB translation, a physical address may be generated via a fixed address mapping system, whereby the upper 3 bits of the virtual address is fixed to 0. Due to this, Pl and P2 are assumed respectively to be 0.5GB, the same as the physical address space.

The reason why Pl and P2 are assumed as the fixed address mapping system is described below. In the example of Fig. 7, it has been described above that the area of P0 is subjected to address translation by the TLB. Herein, it is presumed that a TLB miss has occurred, which indicates the fact that desired address translation information has not been registered in the TLB. In this case, the MMU generates a TLB miss exception, and the OS registers new address translation information in the TLB. The program that performs the TLB replacement is generally stored in a particular area where the virtual address is fixedly mapped in the physical address without being subjected to address translation. As an example, referring to Fig. 7, the program that performs the TLB replacement may be allocated to area P2 of the virtual address. The reason of the above is as described below. If it is adapted so that areas Pl and P2 are also subjected to the address translation by the TLB, when a TLB miss has occurred, there is a possibility that the TLB replacement program for processing the TLB miss cannot be accessed due to a new TLB miss. Thus, in the virtual address system, the particular software of the system may require a virtual address area for mapping a fixed address in the physical address.

In the conventional system of Fig. 7, based on an assumption that the physical address space maximum is 0.5GB, the mapping of the virtual address space is determined. Since it is possible to perform a fixed address mapping from the space of Pl or P2 to any address in the physical address space of 0.5GB, there is no limitation on the fixed address mapping. On the other hand, when the physical address space is extended from 0.5GB in the conventional system, since only the 0.5GB within the extended physical address space is the area that allows the fixed address mapping, it is necessary to carry out the fixed address mapping by selecting the area in the design of the hardware. However, as will be discussed in more detail below, since the area requiring the fixed address mapping varies depending on the system, it typically is necessary to customize the hardware of each system. As a result, typically it is difficult to flexibly cope with the extension of the physical address. For example, since the fixed mapping area of Pl and P2 is the hardware that sets the upper 3 bits of the virtual address to 0, the fixed mapping area of Pl and P2 is linked to 0.5GB only at the start of the physical address. Also, although Pl and P2 have a space of 0.5GB respectively, some applications do not need 0.5GB for the fixed address mapping area. As will be discussed in more detail below in connection with embodiments of the present invention, if it is possible to use a page subjected to the fixed address mapping, and a page subjected to the address translation, while appropriately separating the pages from each other, it is possible to use the virtual address space more efficiently.

It is an object of the invention to provide systems and methods for a data processor capable of extending the size of, mainly, the physical address space. Another object of the invention is to provide systems and methods for ensuring that a supervisor program performing TLB replacement does not issue misses to the TLB. Furthermore, it is another object of the present invention to provide a feature allowing the users who may not use the MMU to extend the size of the physical address space.

Summary of the Invention

In accordance with preferred embodiments of the present invention, a data processor preferably includes a central processing unit CPU and an address translation unit ATU that preferably receives a virtual addresses output from the central processing unit and outputs a physical address, wherein the address translation unit preferably includes a first translation lookaside buffer UTLB, a second translation lookaside buffer DTLB, and a control circuit TLB_CTL for selecting one of the first and second translation lookaside buffers and performing address translation in accordance with an area of an address space in the virtual address.

Brief Description of the Drawings

The above objects and other advantages of the present invention will become more apparent by describing in detail the preferred embodiments of the present invention with reference to the attached drawings in which:

Fig. 1 is an exemplary block diagram of a data processor according to a preferred embodiment of the present invention and having a plurality of translation lookaside buffers (TLBs); utilized in accordance with certain embodiments of the present invention;

Fig. 2 is an exemplary configuration diagram of a first translation lookaside buffer, utilized in accordance with certain embodiments of the present invention;

Fig. 3 is an exemplary configuration diagram of a second translation lookaside buffer, utilized in accordance with certain embodiments of the present invention;

Fig. 4 is an exemplary flowchart of physical address generation, utilized in accordance with certain embodiments of the present invention;

Fig. 5 is an exemplary diagram mapping from a 32-bit virtual address space to a 29-bit physical address space, utilized in accordance with certain embodiments of the present invention;

Fig. 6 is an exemplary diagram mapping from a 32-bit virtual address space to a 32-bit physical address space, utilized in accordance with certain embodiments of the present invention; and

Fig. 7 is an exemplary diagram mapping from a 32-bit virtual address space to a 29-bit physical address space, in accordance with the prior art.

Detailed Description of Exemplary Preferred Embodiments

Hereinafter, referring to the accompanying drawings, a description as to a data processor (or information processing system) and the mode of certain aspects of the preferred embodiments of the system will be made. The circuit elements constituting each block of the embodiment preferably are formed on a semiconductor substrate such as single crystal silicon by means of a technique of semiconductor integrated circuit such as well-known CMOS (complementary MOS transistor) and bipolar transistor or the like, though the elements are not particularly limited thereto. Also, the invention preferably may be realized as an IP module (either soft IP or hard IP) as all or a portion of the design data for forming a semiconductor integrated circuit. The IP module preferably is stored in an information-recording medium, and preferably may be subjected to a trade and/or transfer via a telecommunication line, data network or the like.

Fig. 1 illustrates an example of a data processor as an exemplary preferred embodiment of the invention. Fig. 1 is a block diagram of a data processor 1000 illustrating an address translation unit ATU according to the invention. The data processor 1000 preferably is connected to a central processing unit CPU via a logical address bus LAB, and preferably is connected to a memory controller MEM_CTL via a physical address bus PAB. The address translation unit ATU preferably includes a first translation lookaside buffer UTLB, a second translation lookaside buffer DTLB, an address chop circuit CHOP that preferably generates in a fixed manner a physical address from a virtual address, a TLB control circuit TLB_CTL that preferably controls the above, and a physical address control selector SEL. The DTLB, the

TLB_CTL, and the SEL preferably are included in the address translation unit ATU. Preferably, the first translation lookaside buffer UTLB performs address translation with respect to both data access and instruction access. The third translation lookaside buffer OTLB preferably is provided as a translation lookaside buffer for operands. Reference symbol and numerals OTLB, 106, and 107 denote optional component elements for alternative embodiments as indicated with dotted lines in Fig. 1, and the description thereof will be made below.

Taking the memory access operation of the data processor 1000 as an example, a generating flow of the physical address from the virtual address will be described. The central processing unit CPU preferably includes a control unit and an execution unit, which are not shown in the figures. When performing a memory access, a virtual address 100 is issued by the CPU to the address translation unit ATU.

The TLB control circuit TLB_CTL preferably decodes a part of the virtual address 100 and determines whether (A) output signal 101 of the UTLB should be selected as the physical address using the address translation by the first translation lookaside buffer UTLB, (B) output signal 102 of DTLB should be selected as the physical address using the address translation by the second translation lookaside buffer DTLB, or (C) output signal 103 of the CHOP should be selected as physical address using the address chop circuit CHOP, to control the physical address control selector SEL. At this time, the TLB control circuit TLB_CTL also preferably refers to enable information MMU_ENB_REG of the MMU and enable information DTLB_ENB_REG of the DTLB to control the physical address control selector SEL. The generated physical address 104 preferably is sent to the memory control circuit MEM_CTL and the memory access preferably is carried out.

Fig. 2 and Fig. 3 illustrate a further detailed exemplary configuration of the first translation lookaside buffer UTLB and the second translation lookaside buffer DTLB, respectively. The UTLB illustrated in Fig. 2 is a configuration of an exemplary TLB, which outputs the physical address of an entry corresponding to the compared and input virtual address

100 in accordance with the virtual address and the physical address registered via software. In accordance with an exemplary preferred embodiment of the present invention, the configuration of the TLB preferably includes 64 entries; each entry preferably includes approximately 64 bits including fields of a virtual page number (VPN), an address space identifier (ASID), a valid bit (V), a physical page number (PPN), a page size bit (SZ), and others (OTHER). Examples of OTHER include a change between cacheable and non-cacheable in units of page of the TLB, and a change between copy-back and write-through modes.

Fig. 3 illustrates a configuration of the second translation lookaside buffer DTLB in an exemplary TLB, which outputs the physical address of an entry which corresponds to the compared and input virtual address 100 in accordance with the virtual address and the physical address registered via software. Compared to the above-described UTLB, the number of the entries of the DTLB preferably is smaller. In accordance with an exemplary preferred embodiment of the present invention, the DTLB preferably includes 16 entries and each entry preferably includes approximately 20 bits. Each entry preferably includes the fields of a virtual page number (VPN), an valid bit (V), a physical page number (PPN), a page size bit (SZ), and others (OTHER). Compared to the UTLB, since the number of entries is approximately one fourth and the bits of each entry is approximately one third, it is possible to mount the DTLB in an extremely small TLB. The reason of the above is that, since the DTLB performs the mapping in a larger page unit, by adapting the page size handled by the TLBs to be different between the UTLB and the DTLB the number of the entries can be smaller, and furthermore, the field width of the VPN and the PPN included in each entry can be smaller.

The address chop circuit CHOP in Fig. 1, in accordance with an exemplary preferred embodiment of the present invention, preferably outputs virtual address 100 as the physical address 103, and preferably has a function by which unnecessary upper bits generated due to a difference in size between the virtual address space and the physical address space are set to 0. As an example which will be described later with regards to Fig. 5, a virtual address space of

4GB can be mapped to a physical address space of 0.5GB, the CHOP is a module that preferably sets the upper 3 bits of the 32 bits to 0 while the lower 29 bits therefrom preferably are output as the physical address. Generally, in accordance with such embodiments, the described operation of the CHOP is a function inherent in the hardware, i.e., the CHOP maps a virtual address space area fixedly to a physical address space area via the hardware.

Fig. 4 is an exemplary flowchart that determines, based on the virtual address issued from the CPU (e.g., in accordance with the control of TLB control circuit TLB_CTL) in what manner the physical address should be generated, i.e., by the first translation lookaside buffer UTLB, the second translation lookaside buffer DTLB, or the CHOP.

The TLB control circuit TLB_CTL preferably decodes the upper bits of the virtual address output from the CPU, and determines the area (e.g., any one of P0, Pl, and P2 or the like) in the virtual address (S1). If the virtual address 100 is within an area which is subjected to the address translation by the MMU (e.g., within P0 in the example of Fig. 5, which will be described further below), and if the MMU is enabled (e.g., MMU_ENB_REG in the TLB_CTL is set to "enable"), the TLB control circuit TLB_CTL preferably selects the output signal 101 of the UTLB as the physical address 104 (i.e., S2, S3, and S5). If the virtual address 100 is not within an area which is subjected to the address translation by the MMU (e.g., within Pl and P2 in the example of Fig. 5, which will be described further below), and if the DTLB is enabled (e.g., when the DTLB_ENB_REG in the TLB_CTL is set to "enable"), the output signal 102 of the DTLB preferably is selected as the physical address 104 (i.e., S2, S4, and S7). In other cases, the output signal 103 of the CHOP preferably is selected as the physical address 104 (i.e., S6). Note that, immediately after resetting the data processor 1000, when the MMU and the DTLB are not enabled, since no translation information is set in the TLB, a physical address preferably is generated fixedly by the CHOP output.

Fig. 4 illustrates an example in which the physical address preferably is selected via the control of the TLB_CTL after both of the UTLB and the DTLB have carried out the address translation. However, since the TLB_CTL preferably detects which of the UTLB and the DTLB is selected, low power control may be adopted by carrying out such control so that only the necessary TLB is activated. Thus, in alternative embodiments, the TLB_CTL may be coupled to the UTLB and DTLB to selectively activate the UTLB or the DTLB when the output thereof is to be selected to generate the physical address.

Fig. 5 illustrates a 32-bit virtual address space mapped to a 29-bit physical address space as an example in accordance with one exemplary preferred embodiment of the present invention. In the address translation system illustrated in Fig. 5, area P0 of 2G bytes is the area to be subjected to UTLB translation, and the following areas Pl and P2 of 0.5G bytes are areas to be subjected to DTLB translation. Due in part to the second translation lookaside buffer DTLB (which is provided in accordance with the present invention), the address translation of areas Pl and P2 is carried out in accordance with the contents registered in the DTLB without being limited to fixed mapping. Compared to the conventional example illustrated in Fig. 7, in which areas Pl and P2 are mapped fixedly to the physical address space, certain benefits of the present invention are achieved as compared with the conventional approach.

By allowing the logical address of a supervisor program (e.g., address translation miss handling routine), which performs replacement of the TLB, as well as allowing the DTLB entry, which is used to translate the logical address of an address translation table or the like which the supervisor program accesses, to be resident in the physical address, it is possible to avoid TLB misses during the execution of TLB replacement via the supervisor program. That is to say, control of the entries of the DTLB is made so that the physical address, which is stored in the DTLB entries, is not rewritten even when the second translation lookaside buffer issues an address translation miss. For example, a typical address translation table may be allocated with approximately 8 bytes per page; assuming that one page is 4KB, a capacity of 1MB is required for the physical address of 0.5GB. Since the page size of the DTLB is the range from 16MB to 0.5GB, it is possible to allow this amount to be resident in one entry. However, in the case that one page is 4KB, 256 entries are required to allow this amount to be resident, which may not be practical since a large-capacity TLB is required.

As for the parts of areas Pl and P2 of the above-described TLB (which are not used for the replacement), because in certain examples it is not always necessary to avoid any TLB misses, the areas may be subjected to the address translation. In these examples, the remaining entries not allowed to rewrite a physical address that may be stored upon an address translation miss of the DTLB preferably are allowed to rewrite a physical address that may be stored in the entries upon an address translation miss. Accordingly, by utilizing a large page size of the DTLB, it is possible to make the DTLB to perform a function to complement the UTLB, which preferably is able to perform a finer degree of mapping.

In the prior art example illustrated in Fig. 7, area Pl is a cacheable area (which makes use of the cache), while area P2 is a non-cacheable area (which does not use the cache). In this example, both of areas Pl and P2 are subjected to a fixed address mapping. However, in the case of the DTLB in accordance with the present invention, by assigning a cacheable attribute to the OTHER field of the DTLB, it is possible to control the DTLB while preferably distinguishing between a cacheable area and a non-cacheable area (e.g., which does not use the cache) to a finer degree; i.e., in units of pages that are subjected to the address translation by the DTLB.

In Fig. 5, areas P3 and P4 preferably are further included. P3 preferably is an area subjected to address translation by the UTLB, and preferably is handled in the same manner as area P0. On the other hand, area P4 preferably is an area subjected to address translation by the DTLB, and accordingly it preferably is handled in the same manner as areas Pl and P2. In Fig. 5, the relationship of the address mapping of P3 and P4 is omitted for purposes of clarity.

Fig. 6 illustrates mapping from a 32-bit virtual address space to a 32-bit physical address space as an example of other embodiments of the present invention. In the address translation method illustrated in Fig. 6, similarly to the previously described Fig. 5, area P0 of 2G bytes is the area that is subjected to UTLB translation; the following areas Pl and P2 of 0.5G bytes respectively are the areas that are subjected to DTLB translation. Because of the use of the DTLB, it is possible to carry out the mapping from the virtual address spaces of 1G bytes (including areas Pl and P2) to the physical address space of 4G bytes. An example of the mapping of the UTLB and the DTLB is illustrated in Fig. 6. In this example, although the mapping by the UTLB is mapping of a page size of 4KB, it is shown that a larger area mapping (a large area of 128MB, for example, which is equivalent to a quarter of the area of P1, is mapped via one entry) is carried out by the DTLB. In n this physical address space example, various kinds of spaces such as main memory space (e.g., SDRAM or the like), boot ROM space, PCI space, I/O space and coprocessor space or the like are included, and each may be a space of varying size from several kilobytes to several gigabytes. By providing the DTLB with various page sizes and entries of sufficient number to cover the areas of Pl and P2, it is possible to allow all mapping information of the areas of the respective spaces (necessary for fixed mapping) to be resident in the DTLB. Further, in a configuration in which an MMU is not used, it is possible to adopt a system in which area P0 of 2G bytes is mapped to the physical address space of 2G bytes while the areas of 1G bytes (including Pl and P2 areas) are mapped to the physical address space of 2G bytes using the DTLB.

The areas P3 and P4 in Fig. 6 preferably are handled in the same manner as areas P3 and P4 described in conjunction with Fig. 5. For clarity, the corresponding relationship of the address mapping of areas P3 and P4 are not shown in Fig. 6.

In accordance with certain alternative embodiments of the present invention, data processor may be provided that has a configuration that includes a micro translation lookaside buffer for operand OTLB (an example of which is illustrated in Fig. 1 marked with the dotted line). In certain situations when it may be difficult to use a UTLB (e.g., due to a problem of electric power or speed), a micro TLB may be used. Although a mode wherein the UTLB and the DTLB have a micro TLB may be the simplest mode, in the example shown in Fig. 1, the OTLB as a micro TLB excludes the need for the micro TLB for the DTLB. That is to say, (1) since the micro TLB is of a small capacity, even when the functions of both the UTLB and the DTLB are incorporated, there preferably are no problems in terms of speed; (2) since the micro TLB is a copy of the UTLB and the DTLB, it is renewed automatically by the hardware when any miss has occurred; and (3) even when the copy of the DTLB does not reside in the OTLB, since the necessary entry can be copied from the DTLB by the hardware, another TLB miss does not occur. Accordingly, even when a copy of the DTLB is held, the OTLB may be a typical TLB as it is, in which preferably a TLB miss is allowed. When a large page size is provided to the DTLB but not to the UTLB, an OTLB in accordance with the present invention, is obtained.

The practical operation of the OTLB shown in Fig. 1 is as described below. As in the case that the OTLB is not used, the central processing unit CPU issues a virtual address 100 to the address translation unit ATU. The first translation lookaside buffer UTLB, the second translation lookaside buffer DTLB, the address chop circuit CHOP, and the micro translation lookaside buffer for operand OTLB, preferably translate the virtual address 100 into a physical address. When the OTLB is hit, the TLB control circuit TLB_CTL preferably selects an OTLB output 106 responding to a hit signal 107. If a miss occurs, the output from the UTLB, the DTLB, or the CHOP preferably is selected as the physical address based in part on the decode result of a part of the virtual address 100. In order to reduce power consumption, only when an OTLB miss has occurred, certain aspects of the present invention make it possible to make one or more of the other translation lookaside buffers operate. In additional variations, one or more of the other translation lookaside buffers is made to operate by decoding the virtual address 100 beforehand. Upon an OTLB miss, in preparation for the next access, the hit entry of the UTLB or the DTLB preferably is copied to the OTLB. At this time, based on an algorithm of an LRU (i.e., the least recently used entry is subjected to the replacement) or the like, an object to be subjected to replacement preferably is selected. When a miss has occurred for both the OLTB and the UTLB, a TLB miss exception preferably is generated. As described above, the miss preferably does not occur for the DTLB.

Although certain embodiments in which the invention is applied to a micro translation lookaside buffer for operand OTLB (e.g., as shown in Fig. 1), it is possible to apply certain aspects of the present invention to a micro translation lookaside buffer for an instruction ITLB. In this case, while responding to an instruction fetch operation of the central processing unit CPU, the ITLB preferably is made to operate selectively, and preferably is allowed to operate address translation processing in parallel with the UTLB and the DTLB. In part because of this, the address translation at the instruction fetch operation preferably can be carried out at a higher speed.

Based on certain features of the embodiments described herein, the effect obtained because of the invention disclosed in this application will be described briefly below.

By using the first translation lookaside buffer UTLB and the second translation lookaside buffer DTLB separately depending on the area of the virtual address (e.g., as shown in Fig. 1), the UTLB preferably undertakes the function of the TLB in the conventional technique while the DTLB preferably undertakes a function that substitutes for the conventional technique where the physical address is generated in a manner of fixed mapping. By programming the DTLB, a supervisor program area (typified by a program for performing replacement of the TLB), preferably can generate the physical address to the TLB in accordance with the registered address translation information without causing any misses. In prior art approaches, since the area other than the area which is subjected to the TLB translation is subjected to the fixed mapping by hardware, it typically is impossible to extend/upgrade the size of the physical address space later without changing the hardware on the processor core. However, in the presently describes techniques using the DTLB according to the invention, it is possible to extend/upgrade the size of the physical address space later without changing the hardware of the processor since it is possible to program the DTLB. Furthermore, it is not always necessary that the first translation lookaside buffer UTLB and the second translation lookaside buffer DTLB handle the same page size. Since the DTLB preferably can perform the mapping of the virtual address space to the physical address space in a larger unit, it is possible to reduce the number of the entries of the DTLB. Accordingly, since it is possible to reduce the number of the address translation bits held by the DTLB by performing the mapping in a larger unit, it is possible to largely reduce the size of the hardware necessary for mounting the DTLB compared to the ordinary TLB, and at the same time, it is possible to generate the physical address at a higher speed.

Also, by stratifying the TLB, even when a configuration having a micro TLB (e.g., with a small number of entries for instruction fetch or operand access), the technique according to the invention is applicable in the same manner as described above. Even when pages with different sizes are handled by the UTLB and the DTLB, preferably it is possible for the micro TLB to hold entries from both of the UTLB and the DTLB in a mixed manner.

By employing a processor core using a DTLB according to certain aspects of the present invention, users who do not use the MMU preferably can also utilize a benefit owing to the extension of the physical address space size.

Furthermore, by providing associated information to the OTHER field of the DTLB, it preferably is possible to perform various controls more flexibly. For example, it is possible to control the switching between cacheable and non-cacheable, between copy-back and write-through, or the like. In certain prior art approaches the page size unit of the DTLB might determine these characteristics in a larger space unit.

A processor core having a 32-bit virtual address space, to which the system of the invention is applied, is a processor core preferably having a large flexibility in the address translation method. In addition to the examples illustrated in Fig. 5 and Fig. 6, it is applicable to a 30-bit-1GB physical address space and a 31-bit-2GB physical address space. As demonstrated in the above description, when developing a system mounted with a processor core, in which an address translation unit according to the invention is adopted, it preferably is possible to determine a suitable size of the physical address space and the address map of the system during a development phase irrespective of the hardware features within the processor core. It is therefore possible to prepare a very flexible processor core.

According to the invention, it is possible to extend the physical address space in a data processor of a virtual memory management system easily.

As will be understood by a person of ordinary skill in the present art, the examples discussed here are representative of the full spirit and scope of the present invention. Additional variations, some of which are described here, incorporate many aspects of the present invention.

Although the invention has been described in conjunction with specific preferred and other embodiments, it is evident that many substitutions, alternatives and variations will be apparent to those skilled in the art in light of the foregoing description. Accordingly, the invention is intended to embrace all of the alternatives and variations that fall within the spirit and scope of the appended claims. For example, it should be understood that, in accordance with the various alternative embodiments described herein, various systems, and uses and methods based on such systems, may be obtained. The various refinements and alternative and additional features also described may be combined to provide additional advantageous combinations and the like in accordance with the present invention. Also as will be understood by those skilled in the art based on the foregoing description, various aspects of the preferred embodiments may be used in various subcombinations to achieve at least certain of the benefits and attributes described herein, and such subcombinations also are within the scope of the present invention. All such refinements, enhancements and further uses of the present invention are within the scope of the present invention.